

Query lanjutan dalam SQL adalah teknik pemrosesan data yang lebih kompleks dibandingkan query dasar seperti SELECT, INSERT, UPDATE, dan DELETE. Query ini memungkinkan pengguna untuk melakukan analisis data yang lebih mendalam, mengelompokkan dan memfilter data, menggabungkan beberapa tabel, serta mengoptimalkan pencarian informasi dalam database.



1. Fungsi Agregasi (COUNT, SUM, AVG, MIN, MAX)

Fungsi agregasi digunakan untuk melakukan perhitungan pada sekumpulan data dalam suatu tabel.

- COUNT: Menghitung jumlah baris dalam hasil query.

```
SELECT COUNT() FROM pelanggan;
```

- SUM: Menjumlahkan nilai dalam kolom numerik.

```
SELECT SUM(harga) FROM produk;
```

- AVG: Menghitung rata-rata nilai dalam suatu kolom.

```
SELECT AVG(umur) FROM karyawan;
```

- MIN: Mencari nilai minimum dalam suatu kolom.

```
SELECT MIN(gaji) FROM karyawan;
```

- MAX: Mencari nilai maksimum dalam suatu kolom.

```
SELECT MAX(harga) FROM produk;
```



2. Grup Data dengan GROUP BY

GROUP BY digunakan untuk mengelompokkan data berdasarkan satu atau lebih kolom.

Contoh:

```
SELECT kategori, COUNT() AS jumlah_produk  
FROM produk  
GROUP BY kategori;
```

Hasilnya akan menampilkan jumlah produk berdasarkan kategori.

3. Filter Grup dengan HAVING

HAVING digunakan untuk memfilter hasil yang sudah dikelompokkan dengan GROUP BY, berbeda dengan WHERE yang memfilter sebelum pengelompokan.

Contoh:

```
SELECT kategori, COUNT() AS jumlah_produk  
FROM produk  
GROUP BY kategori  
HAVING COUNT() > 5;
```

Hanya kategori dengan lebih dari 5 produk yang akan ditampilkan.



4. Subquery (Nested Query)

Subquery adalah query di dalam query lain. Ini berguna untuk mengambil data yang bergantung pada hasil query lain.

Contoh:

Menampilkan produk dengan harga lebih tinggi dari harga rata-rata semua produk.

```
SELECT nama_produk, harga  
FROM produk  
WHERE harga > (SELECT AVG(harga) FROM produk);
```

5. Join (INNER JOIN, LEFT JOIN, RIGHT JOIN, FULL JOIN)

JOIN digunakan untuk menggabungkan data dari dua atau lebih tabel.

- INNER JOIN

Mengembalikan hanya baris yang cocok di kedua tabel.

```
SELECT pelanggan.nama, pesanan.id_pesanan
```

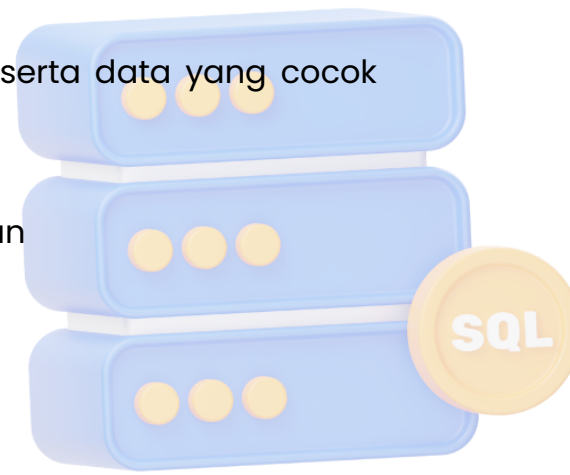
```
FROM pelanggan
```

```
INNER JOIN pesanan ON pelanggan.id_pelanggan =  
pesanan.id_pelanggan;
```

- LEFT JOIN

Mengembalikan semua baris dari tabel kiri, serta data yang cocok dari tabel kanan (jika ada).

```
SELECT pelanggan.nama, pesanan.id_pesanan
```



FROM pelanggan

LEFT JOIN pesanan ON pelanggan.id_pelanggan =
pesanan.id_pelanggan;

- RIGHT JOIN

Mengembalikan semua baris dari tabel kanan, serta data yang cocok dari tabel kiri (jika ada).

SELECT pelanggan.nama, pesanan.id_pesanan

FROM pelanggan

RIGHT JOIN pesanan ON pelanggan.id_pelanggan =
pesanan.id_pelanggan;

- FULL JOIN

Mengembalikan semua data dari kedua tabel, termasuk yang tidak memiliki pasangan.

SELECT pelanggan.nama, pesanan.id_pesanan

FROM pelanggan

FULL JOIN pesanan ON pelanggan.id_pelanggan =
pesanan.id_pelanggan;



6. Union dan Union All

UNION digunakan untuk menggabungkan hasil dari dua atau lebih query dengan struktur yang sama.

- UNION

Menggabungkan hasil query dan menghilangkan duplikat.

```
SELECT nama FROM pelanggan
```

```
UNION
```

```
SELECT nama FROM karyawan;
```

- UNION ALL

Menggabungkan hasil query tanpa menghapus duplikat.

```
SELECT nama FROM pelanggan
```

```
UNION ALL
```

```
SELECT nama FROM karyawan;
```

